(1)

WL-TR-92-1034

DTIC
ELECTE
JUN 1 5 1992
D
C

COMMON DATABASE AIRCRAFT SIMULATION SYSTEM

Uri Bernstein
Seth Greenberg
Bruce Kaye

Technology Service Corporation
2950 31st Street
Santa Monica, CA 90405

May 1992

Report for period Mar 85 - Dec 89

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

AVIONICS DIRECTORATE
WRIGHT LABORATORY
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433-6543

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely Government-related procurement, the United States Government incurs no responsibility or any obligation whatsoever. The fact that the government may have formulated or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication, or otherwise in any manner construed, as licensing the holder, or any other person or corporation; or as conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

John M Woodyard

JOHN M. WOODYARD
CDASS Project Manager
Systems Group

David A Zann

DAVID A. ZANN
Acting Chief
System Integration Branch
Avionics Directorate

FOR THE COMMANDER

Charles H Krueger

CHARLES H. KRUEGER
Chief, System Avionics Division
Avionics Directorate

If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify WL/POOS, WPAFB, OH 45433-6563 to help us maintain a current mailing list.

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
May 1992

3. REPORT TYPE AND DATES COVERED
Final Mar 85 - Dec 89

4. TITLE AND SUBTITLE
Common Database Aircraft Simulation System

5. FUNDING NUMBERS
C-F33615-84-C-1542
PE-62204F
PR-2003
TA-02
WU-72

6. AUTHOR(S)
Uri Bernstein, Seth Greenberg, and Bruce Kaye

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Technology Service Corporation
2950 31st Street
Santa Monica CA 90405

8. PERFORMING ORGANIZATION REPORT NUMBER
TSC-PD-A340-11

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
John M. Woodyard (513) 255-4827
Avionics Directorate (WL/AAAS)
Wright Laboratory
Air Force Systems Command
Wright-Patterson AFB, OH 45433-6543

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
WL-TR-92-1034

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The Common Database Aircraft Simulation System (CDASS) is a three-channel flight simulator support system which generates real-time cockpit imagery for visual, infrared (IR), and radar sensors. CDASS implements two significant advances in cockpit simulation. First, it provides three-channel simulation with low-cost off-the-shelf processor and imaging technology. Second, CDASS is based on the concept of a common simulation database which is a unified representation of the simulation scene containing multiple sensor information. The common database guarantees that correlated images will be displayed in all the simulation channels.

14. SUBJECT TERMS
Aircraft Simulator, Common Database, sensor simulation

15. NUMBER OF PAGES
53

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev 2-89)
Prescribed by ANSI Std Z39-18
298-102

FOREWORD

This Final Technical Report is submitted as a requirement of the Common Database Aircraft Simulation System (CDASS), Contract Number F33615-84-C-1542. This document is the final submission for Contract Data Requirements List (CDRL) Sequence Number 14, Attachment No. 1, Technical Reports/Final Technical Report.

This report was written by Uri Bernstein, Seth Greenberg, and Bruce Kaye.

CONTENTS

CONTENTS (CONT'D)

CONTENTS (CONT'D)

LIST OF FIGURES

SYMBOLS AND ACRONYMS

ASCII	American Standard Code for Information Interchange
AVSAIL	Avionics System Analysis and Integration Laboratory
CAD	Computer-Aided Design
CDASS	Common Database Aircraft Simulation System
CDB	Common Data Base
CDRL	Contract Data Requirements List
CIG	Computer Image Generation
DEC	Digital Equipment Corporation
DMA	Defense Mapping Agency
DTED	Digitized Terrain Elevation Data
FLIR	Forward-Looking Infrared
FOV	Field Of View
GSL	Graphics Subroutine Library
Hz	Hertz
IR	Infrared
nmi	nautical miles
PHIGS	Programmer's Hierarchical Interactive Graphics System
RBGM	Real Beam Ground Map
RGB	red, green, blue
SAR	Synthetic Aperture Radar
TSC	Technology Service Corporation

VDASS	Videodisc Aircraft Simulation System
VLT	Video Look-Up Table
VMS	Virtual Memory System
WRDC	Wright Research and Development Center

1.0 INTRODUCTION

1.1 PURPOSE

The Common Database Aircraft Simulation System (CDASS) is a three-channel flight simulator which generates real-time cockpit imagery for visual, infrared (IR), and radar sensors. CDASS was developed by Technology Service Corporation (TSC) under contract to Wright Research and Development Center (WRDC), and is now installed and operating at WRDC's Avionics System Analysis and Integration Laboratory (AVSAIL).

This document, the Final Technical Report for CDASS, describes the design objectives and the major design features of the system. It is intended to provide an overall understanding of the entire CDASS system and its underlying concepts.

1.2 SCOPE

This report assumes no familiarity with the CDASS system, and is therefore the appropriate introduction to CDASS. It contains a comprehensive top-level description of CDASS, including the hardware, common data base (CDB), and the real-time and off-line algorithms and software.

Several additional CDASS documents provide more detailed information on various aspects of the system. The *User's Manual* contains detailed operating instruction; the Computer Program Product Specification describes the real-time and off-line software; the Software Maintenance Manual provides instructions for the maintenance and modification of the real-time on off-line software.

1.3 ORGANIZATION AND CONTENT

Section 2 describes the objectives and history of CDASS, while Section 3 summarizes the actual system design. The CDB and its associated off-line programs are outlined in Section 4. Section 5 describes the operation of the real-time visual and forward-looking infrared (FLIR) channels, and Section 6 describes the synthetic aperture radar (SAR) channel. Section 7 describes the external and internal real-time operating modes of the system. Section 8 presents the conclusions drawn from the CDASS development effort.

2.0 OBJECTIVES AND HISTORY

2.1 INTRODUCTION

Real-time aircraft cockpit simulation is an increasingly important technology for personnel training, human factors research, and aircraft sensor and display development. Applications of cockpit simulation have grown rapidly in recent years due to the increased capabilities for real-time imaging simulation and to the decreased cost of the technology.

CDASS is a three-channel simulator which implements two significant advances in cockpit simulation. First, it provides three-channel simulation with low-cost off-the-shelf processor and imaging technology. Second, and perhaps most important, CDASS is based on the concept of a common simulation data base which is a unified representation of the simulation scene containing multiple sensor information. The common data base guarantees that correlated images will be displayed in all the simulator channels.

2.2 OBJECTIVES

The principal objectives of CDASS were stated in Section C of the Statement of Word for F33615-84-C-1542:

"... to develop a low cost, multi-channel Computer Image Generation (CIG) system which operates from sensor specific databases derived from a single generic database. The system will provide real-time, fully interactive, man-in-the-loop visual and sensor displays that correspond to similar displays in real aircraft under actual flight conditions. The system shall be capable of both stand-alone operation using an integral airframe model or capable of supporting a remote host simulation system as a correlated sensor visual subsystem."

The performance requirements of CDASS are as follows:

1. Out-the-window display, pseudo-synthetic aperture radar (SAR) display, and pseudo-forward looking infrared (FLIR) display.
2. Unrestricted six-degree-of-freedom flight over a minimum 20 by 50 nmi gaming area.
3. Realistic changes in display in accordance with changes in aircraft attitude and position.
4. Aircraft velocity of 0 to 500 knots.

5. **Gaming area to contain sufficient cultural features to provide for realistic mission scenarios.**
6. Operate in either a independent mode, or a mode in which the simulator is driven by the AVSAIL Harris computer.
7. Allow for future creation of CDASS data bases from Defense Mapping Agency (DMA) data.

2.3 HISTORY

The current CDASS development effort began with the Videodisc Aircraft Simulation System (VDASS) program in March 1985. The primary objective of VDASS was low-cost, real-time, six-degree-of-freedom simulation using videodisc technology. Three-channel data was to be recorded off-line on videodisc at selected positions and platform orientations with respect to the data base; at simulation run-time, the displayed image was to be interpolated in real time from the recorded data.

In the initial phase of the VDASS program, TSC showed that the video-disc state-of-the-art was not capable of supporting fully interactive, 30 Hz, high-fidelity three-channel image generation. An alternative based on computer image generation (CIG) was proposed and VDASS became the CDASS program.

The common data base concept and the data base development environment were added to the program. Work on CDASS continued from 1986 through the beginning of 1988, when the simulation hardware and software were completed. The data base development software and documentation were completed in 1989. The system was installed at WRDC's AVSAIL facility in September 1989.

3.0 CDASS SYSTEM SUMMARY

3.1 INTRODUCTION

CDASS is a complex system consisting of multiple hardware and software systems. Functionally, it can be divided into two distinct systems: simulation and data base development. The simulation (on-line) system provides the three-channel real-time flight simulation for a selected data base. The data base development system (off-line) provides the tools for building data bases which can be used by the simulation system. These two functional systems have their own software systems, which both run on the CDASS hardware resources.

3.2 SIMULATION SYSTEM

The simulation system performs the basic flight simulation functions. These functions consist of:

1. Initialization and data base loading.
2. Interface to the AVSAIL host simulation computer, while operating in host mode. In this mode, CDASS accepts aircraft position and orientation updates from the AVSAIL host.
3. Pilot interface management and aircraft motion simulation, in stand-alone mode. In this mode, CDASS generates aircraft position and orientation updates by reading the joysticks and simulating aircraft motion.
4. User interface for freezing, resetting, and resuming CDASS operation.
5. Visual channel display update
6. FLIR channel display update
7. SAR interface management. The user can select the orientation, location, and display scale for the SAR patch map display.
8. SAR shadow data update. The SAR shadows depend on the aircraft position with respect to the data base, and must be updated whenever a SAR image is generated from a new aircraft position.
9. SAR channel display update

3.3 DATA BASE DEVELOPMENT SYSTEM

The data base development system allows the CDASS user to build and modify CDASS data bases, starting from the basic specifications of individual data

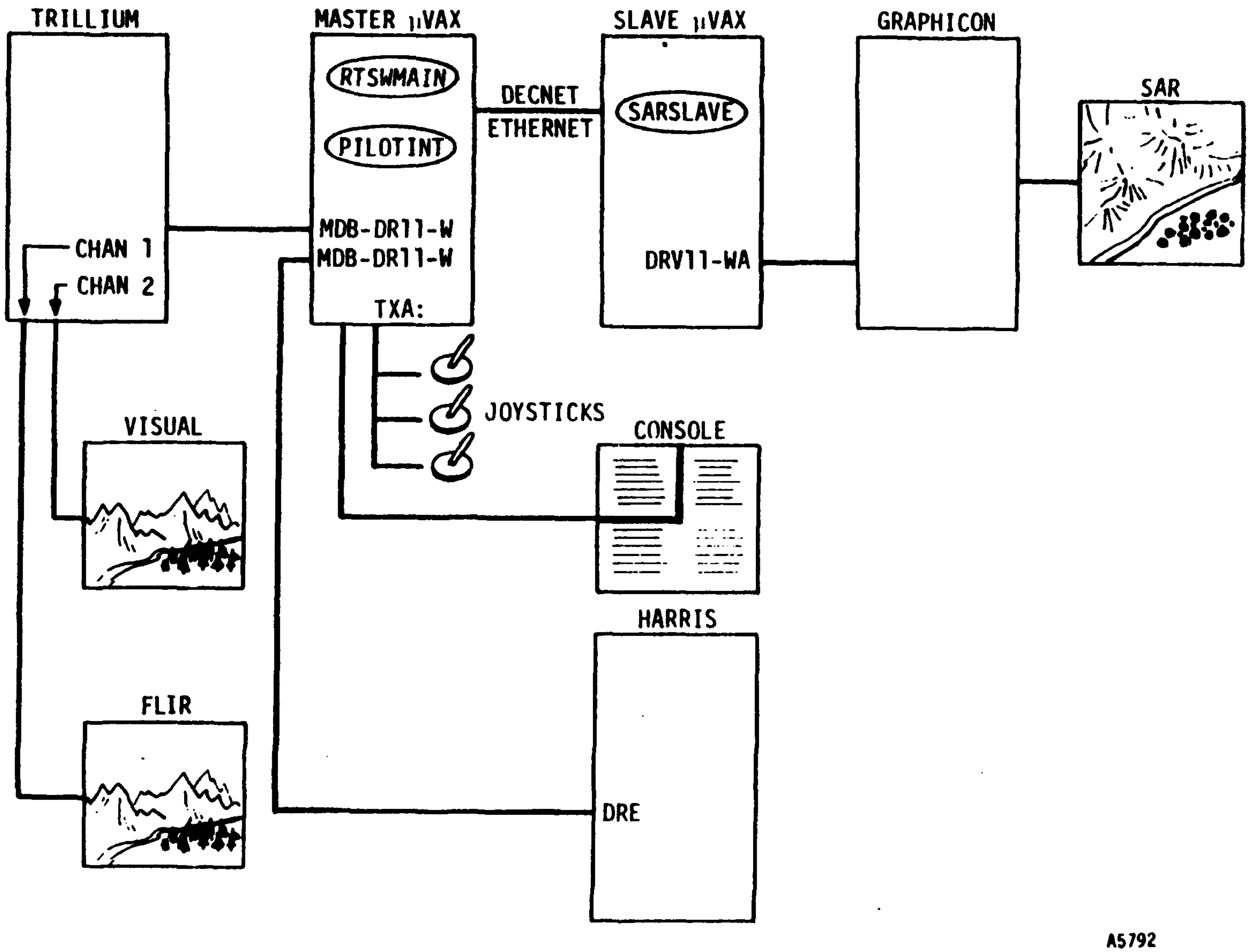
base objects. The data base development system performs three major functions:

1. Creation of common data base objects (vehicles, structures, terrain features) from basic geometric information. The data base objects become part of the CDB object library.
2. Creation of a common data base scenario by placement of the data base library objects within a scene. CDB scenarios (called world files) can be stored in a scenario library.
3. Creation of the three individual channel data bases from the common data base. These individual data bases are loaded by the simulation system.

3.4 HARDWARE

Diagrams of the CDASS hardware are shown in Figures 1 and 2. The hardware consists of the following major components:

1. Two (2) Digital Equipment Corporation (DEC) MicroVAX II general-purpose minicomputers, connected by an ethernet local area network. These two computers are referred to as the master and slave microVAXes.
2. Trillium Corporation model 1102 two-channel computer image generator, connected to the master microVAX by a DR-11W high-speed parallel interface.
3. Star Technologies (formerly General Electric) Graphicon 1700 graphics processor, connected to the slave microVAX by a DRV-11WA high-speed parallel interface.
4. Three (3) high-resolution display monitors (two Barco CD-351 for the visual and FLIR displays and one Hitachi for the SAR display).
5. A PC-compatible microcomputer, connected to the master microVAX by an RS-232 interface.
6. Three (3) alphanumeric display terminals (two DEC VT-220 and one Espirit 10/102).
7. Three (3) KA Design joystick.



OVERVIEW OF RUNNING SYSTEM

FIGURE 1. OVERVIEW OF CDASS RUN-TIME SYSTEM

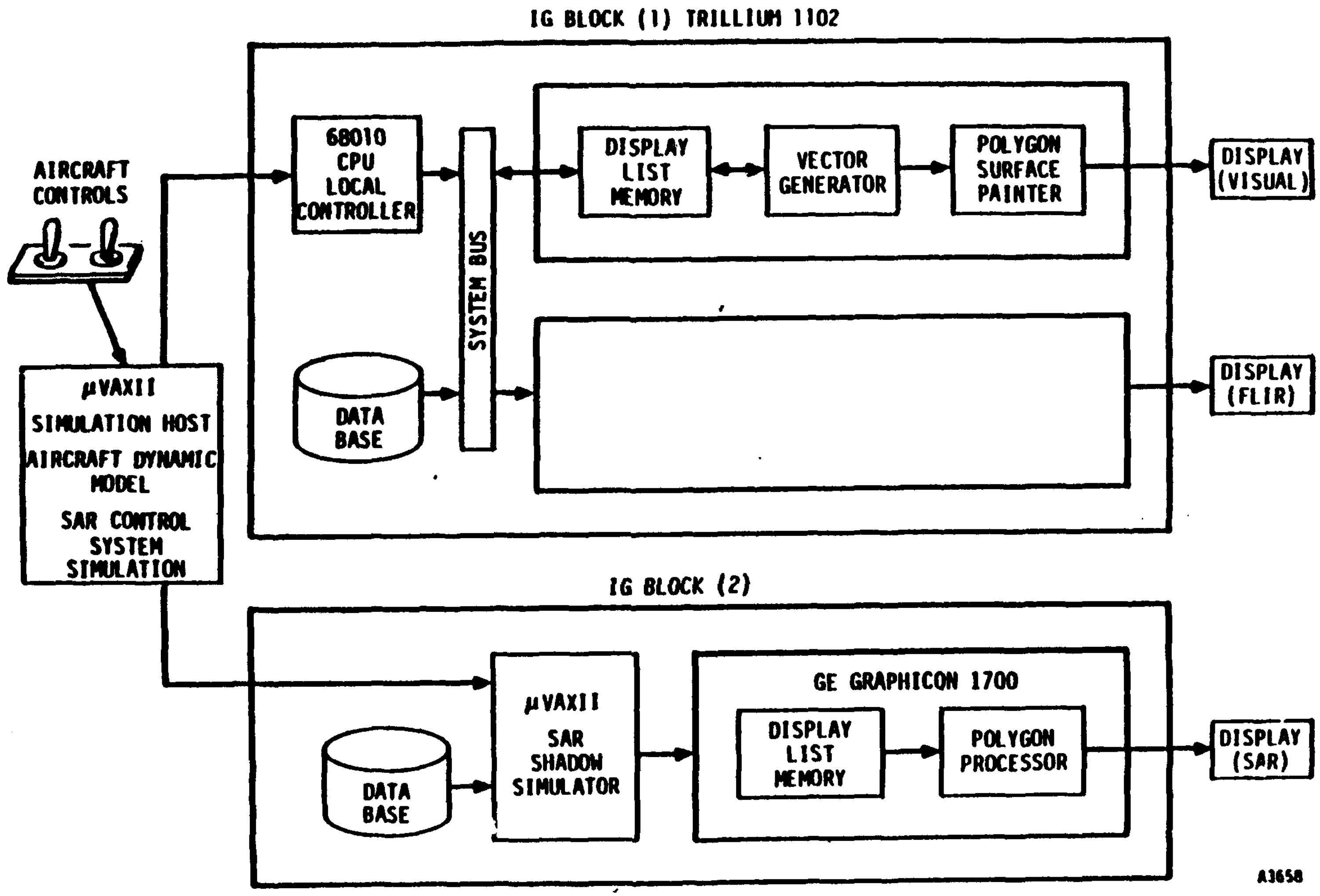


FIGURE 2. CDASS SYSTEM BLOCK DIAGRAM

3.5 SOFTWARE

A data flow diagram for the real-time software is shown in Figure 3. The CDASS real-time simulation software consists of the following components:

1. The RTSWMAIN process runs on the master microVAX. This process reads the host computer interface, reads the pilot and SAR joysticks, simulates aircraft motion (stand-alone mode), receives aircraft state information (host mode), updates the console display, interfaces with the Trillium, and communicates with the SARSLAVE process.
2. The SARSLAVE process runs on the slave microVAX. This process computes the shadows for the SAR display, loads the Graphicon, and updates with the Graphicon. A data flow diagram for the slave microVAX software is shown in Figure 4.
3. The PILOTINT process runs on the master microVAX. This process manages the user console interface.

The CDASS off-line data base development software and its interfaces are shown in Figure 5. This software consists of the following components:

1. AutoCAD Release 10, which runs on the PC, generates object geometric data files with visual, FLIR, and SAR color attributes.
2. The data base conversion program runs on the slave microVAX. It translates AutoCAD DXF files to CDB files, and adds the three channel information to the data base object.
3. The world editor, which runs on the slave microVAX, creates and edits CDB world files (scenarios) from the object library.
4. The filter program converts CDB files into the three channel data bases which are derived from the parent CDB file. The filtering program creates three individual data bases in the CDB format.

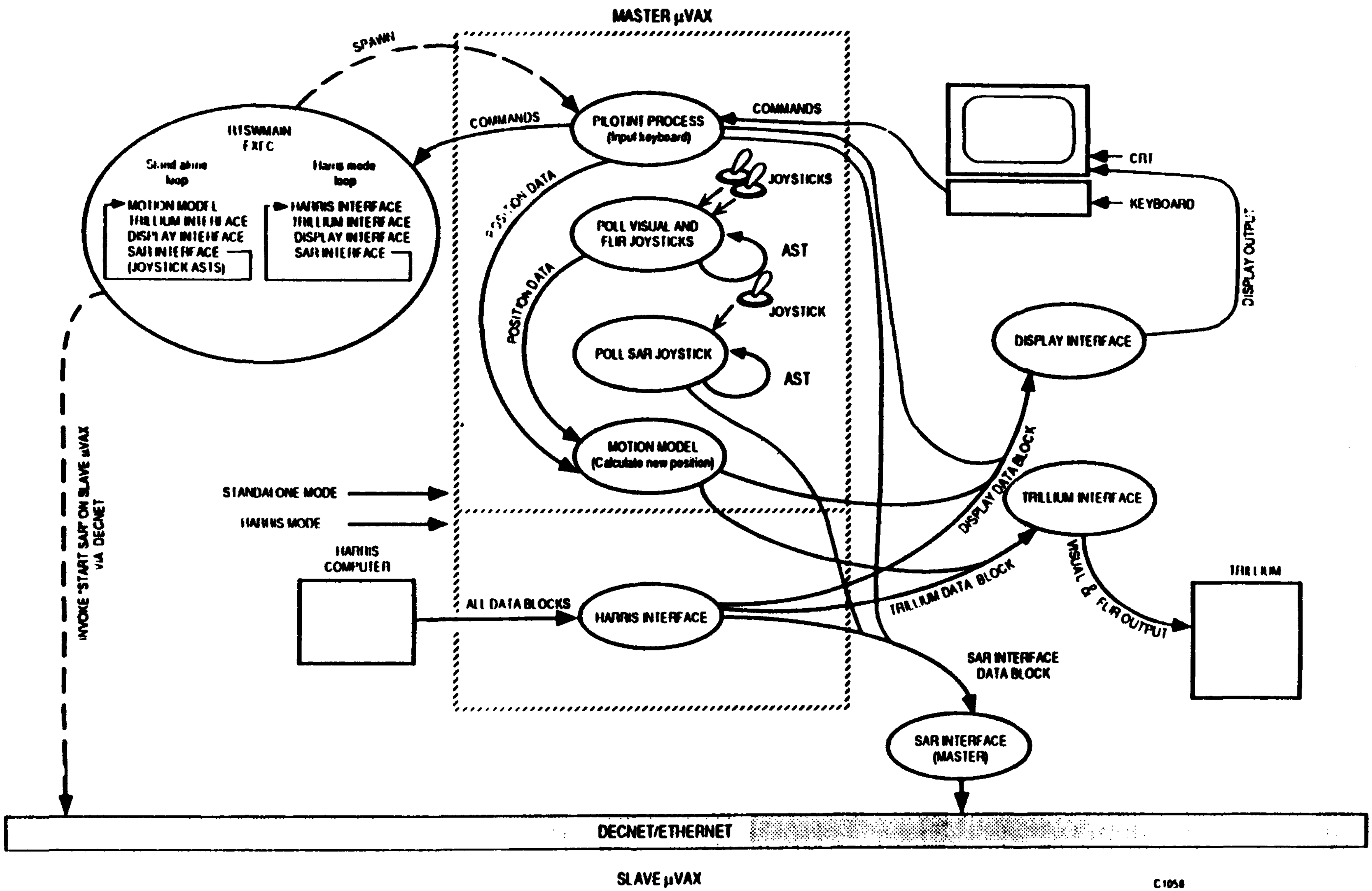


FIGURE 3. CDASS REAL-TIME SOFTWARE DATA FLOW (PART 1)

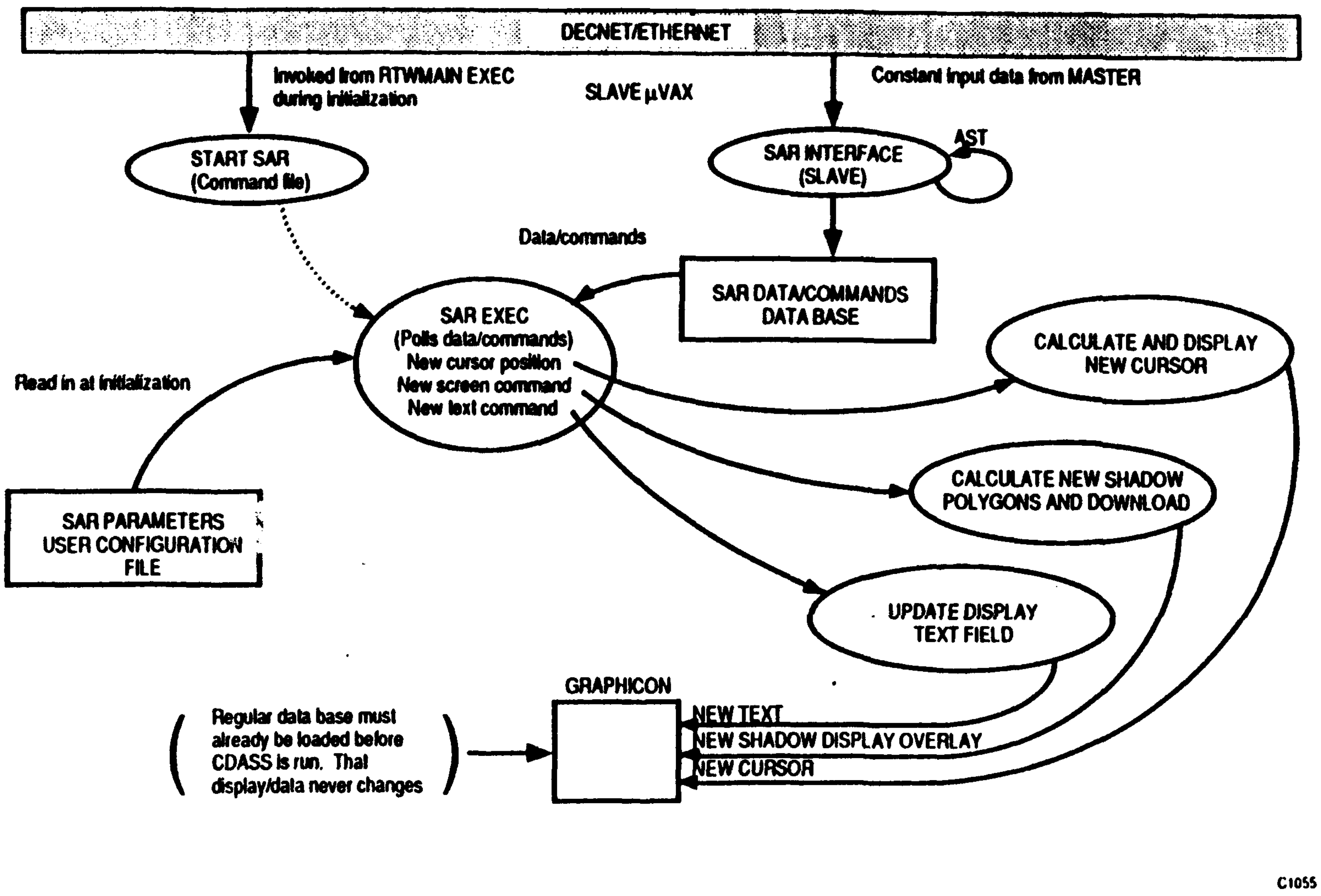


FIGURE 4. CDASS REAL-TIME SOFTWARE DATA FLOW (PART 2)

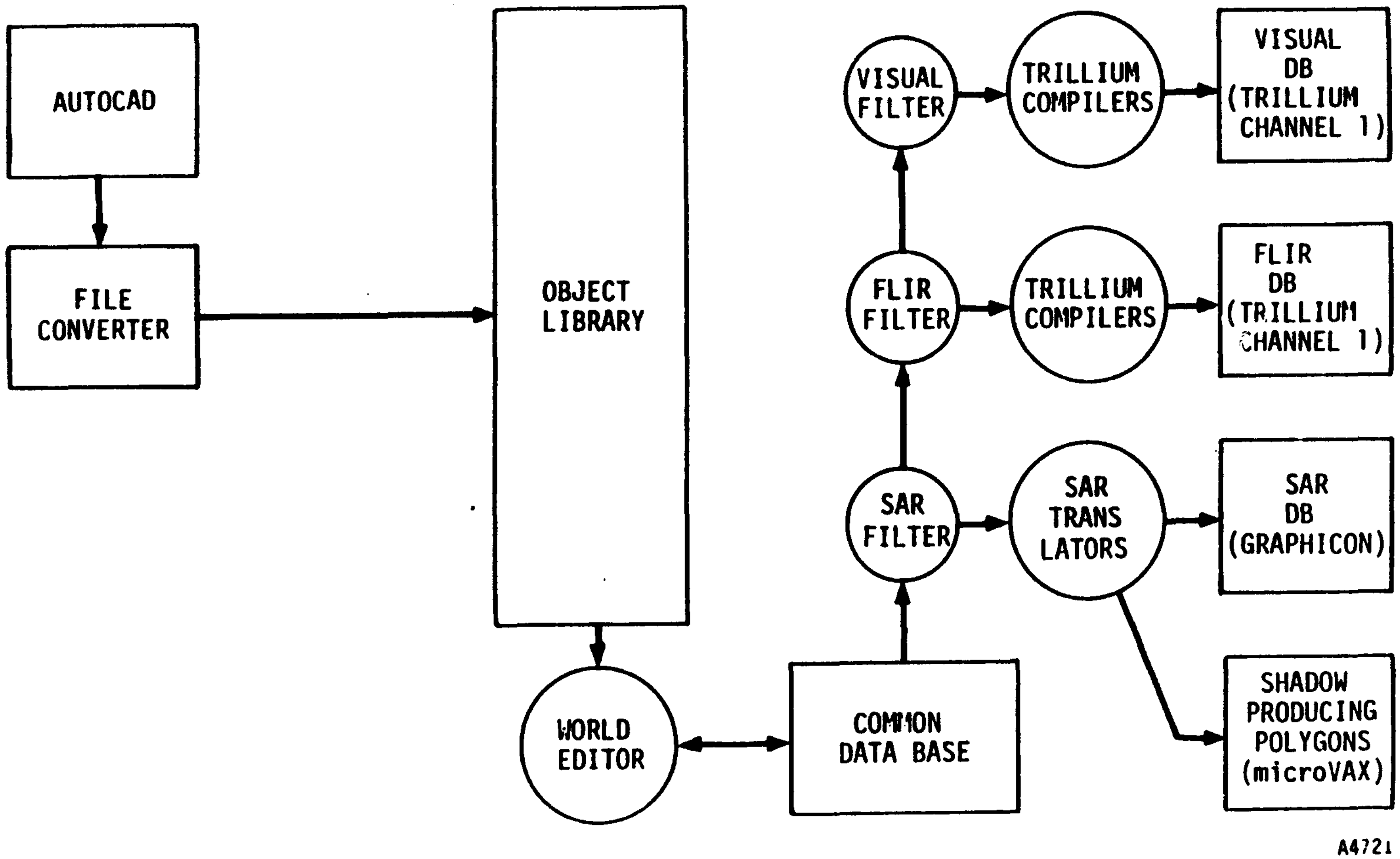


FIGURE 5. CDASS DATA SYSTEM

5. The Trillium compiles the visual and FLIR CDB data bases into format that can be loaded into the Trillium.
6. The Graphicon translation programs convert the SAR CDB data base into a format that can be loaded into the Graphicon. They also create the shadow polygon data base from the original data base.

4.0 THE COMMON DATA BASE

4.1 INTRODUCTION

The Common Data Base (CDB) is the key to the CDASS system. It describes the simulation world by specifying both its geometric properties and its signatures for visible, IR, and radar sensors. The design of the CDB and the CDASS simulator guarantee that multi-channel simulations will provide an accurately correlated view of any scenario specified in the CDB format.

The CDB was designed to be a hierarchical, device-independent graphics data base containing both geometric and three-channel signature information. Its design was derived from the Trillium data base, which in turn was derived from the Programmer's Hierarchical Interactive Graphics System (PHIGS), an industry standard. The CDB describes objects in terms of a surface polygon representation, since this is the representation that is used by most computer image generation system.

This section describes the CDB and its constituent files, and the CDASS programs which create and modify the CDB.

4.2 COMMON DATA BASE STRUCTURE

The CDB contains three types of files. In descending hierarchical order, these are the world file, the object file, and the atom file. The atom file describes a basic geometric entity and specifies its coordinates. The object file describes a related group of atoms or objects and includes their relative geometric placement as well as the three-channel signature data. The world file describes a complete CDASS scenario and includes objects and information about their placement within the scenario world. Different world files may contain the same objects.

One of the key features of the CDB is the use of instancing. Instancing is the capability to create several distinct copies (instances) of the same graphic entity. For example, a model of a tank only has to be specified once in the CDB. An arbitrary number of copies of that tank can be inserted into a scenario by simply referencing the original tank object file.

World files are created and modified by the World Editor program. Object and atom files are created by the file conversion program from AutoCAD files. Any of these files can also be created manually using an editor and following the file formats given in the CDASS Computer Program Product Specification.

All CDB files are ASCII files, named in accordance with the VAX Virtual Memory System (VMS) file naming rules. Each file has an three-letter extension which identifies whether it is a world, object, or atom file.

4.2.1 The World File

The world file (file extension .WLD) contains the information describing how objects are placed in the data base with respect to one another. For each object, an (X,Y,Z) location is specified within the world coordinate system. In addition to location, scaling and rotation transformations may be applied to objects in a world file. The following is an example of a world file which contains three instances of the same "house" object.

```
WORLD
   BLK
     HOUSE
     HOUSE M(50,50,0),RZ(.25,0,0)
   END
   BLK
     HOUSE M(1150,1250,0)
   END
END
```

As can be seen in this example, the world file may be separated into a number of blocks, which are groups of objects within the same neighborhood. Each block defines a separate bounding box which speeds the processing of the Trillium display.

4.2.2 The Object File

The objects referenced in the world file are described in object files (file extension .CDB). (The .OBJ extension is already used for VMS relocatable object and Trillium object files.) Within an object file, the constituent atoms or objects are organized by level of detail.

For each level of detail, the object is described by one or more components that make up the total object. These components may themselves be other objects or may be atoms. It is at this level that the atoms are assigned a visible, IR, and radar signature. The atoms and subobjects may also have geometric transformations applied to them, consisting of translation, scaling, and rotation about any or all of the three axes. The following is an example of an object file, containing two instances of the atom "panel" and one instance of the atom "square."

```
VFS OBJECT ROOF
VFS   LEVEL01 200000
VFS     ATOM PANEL
VFS     TRANS RX(0.125,0.0,0.0)
VFS     TRANS M(0.0,0.0,1.0)
V       COLOR (125,125,125)
 F      COLOR (511,511,511)
  S     COLOR 16
VFS     ;
VFS     ATOM PANEL
VFS     TRANS RX(-0.125,1.0,0.0)
VFS     TRANS M(0.0,.415,1.0)
V       COLOR (125,125,125)
 F      COLOR (511,511,511)
  S     COLOR 16
VFS     ;
VFS   LEVEL02 2000
```

```
VFS     ATOM SQUARE
V       COLOR (125,125,125)
 F      COLOR (511,511,511)
  S     COLOR 16
VFS     ;
VFS END
```

Every line in the object file must begin with a channel designator. This is a three-character string in which the first character indicates the presence or absence of the object in the visible (V) channel, the second character indicates the presence or absence of the object in FLIR (F) channel, and the third character indicates the presence or absence of the object in the SAR (S) channel. The signature for each channel follows the keyword "COLOR" in the object file. Visual signatures are RGB (red, green, blue) color triplets; FLIR signatures are gray shades (duplicated as RGB triplets); radar signatures are an index into a radar reflectivity file, which will be discussed in Section 6.

The independent channel assignments allow certain surfaces to be active in one channel and not in another. One example is a "hot spot" which would have a different gray shade in the IR from its surrounding surface, but would not appear in the visual or SAR channels. A second example would be a corner reflector, which would have a very strong return in the SAR channel, and could be negligible in the visual and IR channels.

The level of detail (not implemented in the current version of CDASS), allows the data base creator to specify varying level of detail as a function of viewing range. At long range, where resolution is low and many objects are in the field of view, a low level of detail is used, reducing the total polygon count and allowing the many objects in the field to be imaged in real time. Conversely, at short ranges, where fewer objects are visible, a high-resolution model can be used without slowing the image generation.

4.2.3 The Atom File

The atom is the basic building block of all objects. It consists of one or more planar triangles, which are specified by their vertex coordinates and their normal vectors. Curvilinear surfaces must be represented by a faceted triangular approximation. (The Trillium and Graphicon can both accept polygons with more than three vertices, but the CDB atoms are limited to triangular polygons to assure complete device independence.)

Atoms are described in the atom file (file extension .ATM). Each atom is named and defined in two parts. The first part lists the points and normals of the atom, and the second part specifies the polygons which constitute the atom and the normals for each polygon.

The following is an example of an atom file describing a horizontal triangle.

```
ATOM PANEL
   POINT0001      0.0000      0.0000      0.0000
   POINT0002      1.0000      0.0000      0.0000
   POINT0003      1.0000      1.0000      0.0000
   NORMAL0001     0.0000      0.0000      1.0000
SURFACE<
   POLYGON0001    1    2    3
   VN(  1,     1 )
   VN(  2,     1 )
   VN(  3,     1 )
>
END
```

The first section of the atom file specifies three points by their x, y, and z coordinates. It also specifies one normal vector. The second section defines one surface which is connects points 1, 2, and 3, and three vertex normal vectors which are all identical in this example. A different normal vector could be defined at each vertex to allow Gouraud or Phong shading of polygonal segments of curved surfaces.

Polygons follow the right hand rule for visibility. If the point listing in the POLYGON statement is in counter-clockwise order (as in the above

example), the polygon surface is visible from the top. If the listing is in clockwise order, the polygon surface is visible from the bottom.

4.3 THE OBJECT MODELER

The object modeler is a set of two programs which allows the user to create the atom and object constituents of the CDB library. Originally, the object modeler and world editor were conceived as a single off-the-shelf interactive graphics design tool, which would run directly on the Trillium or Graphicon. However, no such commercial product was available for either machine during the development phase of CDASS. Therefore, separate object modeler and world editor programs were implemented.

The object modeler consists of two programs. The first is AutoCAD Release 10, running on an PC. This off-the-shelf program creates object data files which are transmitted to the slave microVAX. There, a specially written file conversion program transforms the AutoCAD files into CDB atom and object files. The result is a system which automatically converts an AutoCAD design into the appropriate CDB library files.

AutoCAD was chosen because of the full 3-D capabilities of Release 10, because of its openly available file formats, and because of its position as an established and proven computer-aided design (CAD) program.

4.3.1 Object Creation

The object modeler is capable of handling all the AutoCAD 3-D entities, which include the following:

- 3-D face
- 3-D closed polyline
- 3-D curved mesh
- explicit 3-D objects (box, cone, dome, sphere, etc.)

A single AutoCAD drawing may contain several of these entities. The user must save the drawing as a DXF file, and must send the file to the slave microVAX. The conversion program then transforms the DXF file into one CDB object file. Each entity within the drawing becomes an individual CDB atom file.

4.3.2 Three-Channel Signature Assignment

As part of the object creation process, each atom is assigned a three-channel signature, as described in section 4.2.2. The assignment is performed automatically by having the user assign an AutoCAD color index to each entity within the AutoCAD drawing. When the files are converted to the CDB format, the index is used as a lookup into a surface properties table. A small section of such a table is given as an example below:

1	'VF '	450	412	505	234	0	wood, white paint
2	'VFS'	199	215	281	326	3	gray shingle roof
3	'VFS'	128	470	50	0	29	steel, olive green paint
4	' S'	0	0	0	0	25	wire fence

4.4 THE WORLD EDITOR

The world editor is an interactive graphics program which runs on the slave microVAX and Graphicon. It creates and modifies world files, using object files from the CDB library. The world editor is controlled from the slave microVAX terminal keyboard and by a graphics tablet and knob box connected to the Graphicon.

The world editor performs the following functions:

1. **Adding an Object to the World.** The user selects an object from the library and specifies the point in the world at which it is to be inserted.

2. **Deleting an Object from the World.** The user can select an object in the world and delete it. The object to be deleted can be specified by its sequence number or by interactive selection with the tablet.

3. **View Control.** This function allows the user to specify the viewing (eye) position and the direction of look. It also allows the user to move about the data base in specified increments. A split-screen multiple-view option can also be selected.

4. **Object Control.** This function allows an existing object in the world to be translated, rotated, or scaled.

5. **Reading an Existing World File.** This function is used to modify an existing world file.

6. **Writing a World File.** This function is used to save a world file that has been created or modified.

The current world editor displays only the visual channel data base. The displayed colors, however, do not correspond to the colors seen on the Trillium display in the real-time simulation. This display inconsistency is due to the fact that the Trillium has a 27-color plane and can display 134 million colors simultaneously, while the Graphicon can only display 4096 colors simultaneously. Therefore, the world editor must map the 27 bits of Trillium color into 12 bits of Graphicon color.

4.5 CDB FILE FILTERING AND TRANSLATION

To produce a run-time graphic data base for CDASS, a CDB data base must be separated into three separate sensor channel data bases. The CDB filter program is used to accomplish this. The filter reads through the CDB object files, determining which lines belong in which sensor's data base, and writes

each line into the appropriate channel data base. The outputs of the filter are the three "child" data bases, one for each channel.

Since the CDB and Trillium data base formats are very similar, the Trillium atom, object and world compilers can directly process the visible and FLIR filtered channel outputs. Therefore, the filtered visual and FLIR channels do not require any further processing by the CDASS software. (See the next section for the relationship between the CDB and Trillium data bases.) However, the SAR data base requires additional translation, because the Graphicon's graphics specification language is very different from the CDB format. Therefore, there is an additional translation program (TRIL2GRAPH.EXE) which converts the filtered SAR data base into Graphicon's Graphics Subroutine Library (GSL) commands.

The details of the Trillium and Graphicon data base representation are described in Sections 5 and 6.

4.6 THE TERRAIN BELT DATA BASE

One of the older flight simulation techniques used a video camera suspended over a model of the simulation scene. Platform motion was simulated by moving the camera, terrain model, or both. Such a simulator has been operating at AVSAIL, using a specially created scene known as the terrain belt.

One of the CDASS project deliverables was a CDB model of the AVSAIL terrain belt. The model contains all the major topographical features of the terrain belt, including several mountain ranges, a river, and a lake. It also includes many cultural features such as two airfields, roads, and a town. The CDASS terrain belt can be used for three-channel simulation, while the old terrain belt model was restricted to the visual channel.

4.7 DMA CONVERSION

One objective of CDASS was to create a simulation and data base that could, in the future, use DMA data as an input to the system. The Trillium Corporation provided a limited DMA conversion capability as part of the 1102 system delivery. This conversion program reads DMA digitized terrain elevation data (DTED) and creates compiled Trillium files. This program could not be readily modified to create CDB files, and cannot be used as part of CDASS. Furthermore, TSC's tests of Trillium's program in the visual channel showed several problems in creating a visually convincing terrain image.

5.0 THE VISUAL AND FLIR CHANNELS

5.1 INTRODUCTION

The visual and FLIR channels of CDASS are essentially identical in processing. Both are displayed on independent channels of the Trillium, using their respective data bases derived from the CDB.

5.2 THE TRILLIUM GRAPHICS GENERATOR

The Trillium Model 1102 is a two-channel 3-D image genera'~r, intended to be used with a VAX host. In 1986, when it was selected for C 'SS, it represented a breakthrough in simulation performance at its price level. In addition to responding to commands from the host VAX, the Trillium can operate in an independent flight simulation mode, controlled by its own joysticks. This latter capability is not used by CDASS. The Trillium has the following features:

a. Display capacity of up to 12,500 edges per frame at 30 Hz
b. Smooth shading
c. Movable light sources
d. 640-by-480 resolution
e. Fog/haze option (not used in CDASS)
f. Moving objects

A system block diagram of the Trillium is shown in Figure 6.

The Trillium has two removable cartridge disk drives which give it a data base storage capability independent of the VAX host. The cartridge tapes have a capacity of 20 Mbytes and are used to store the Trillium versions of the channel data bases.

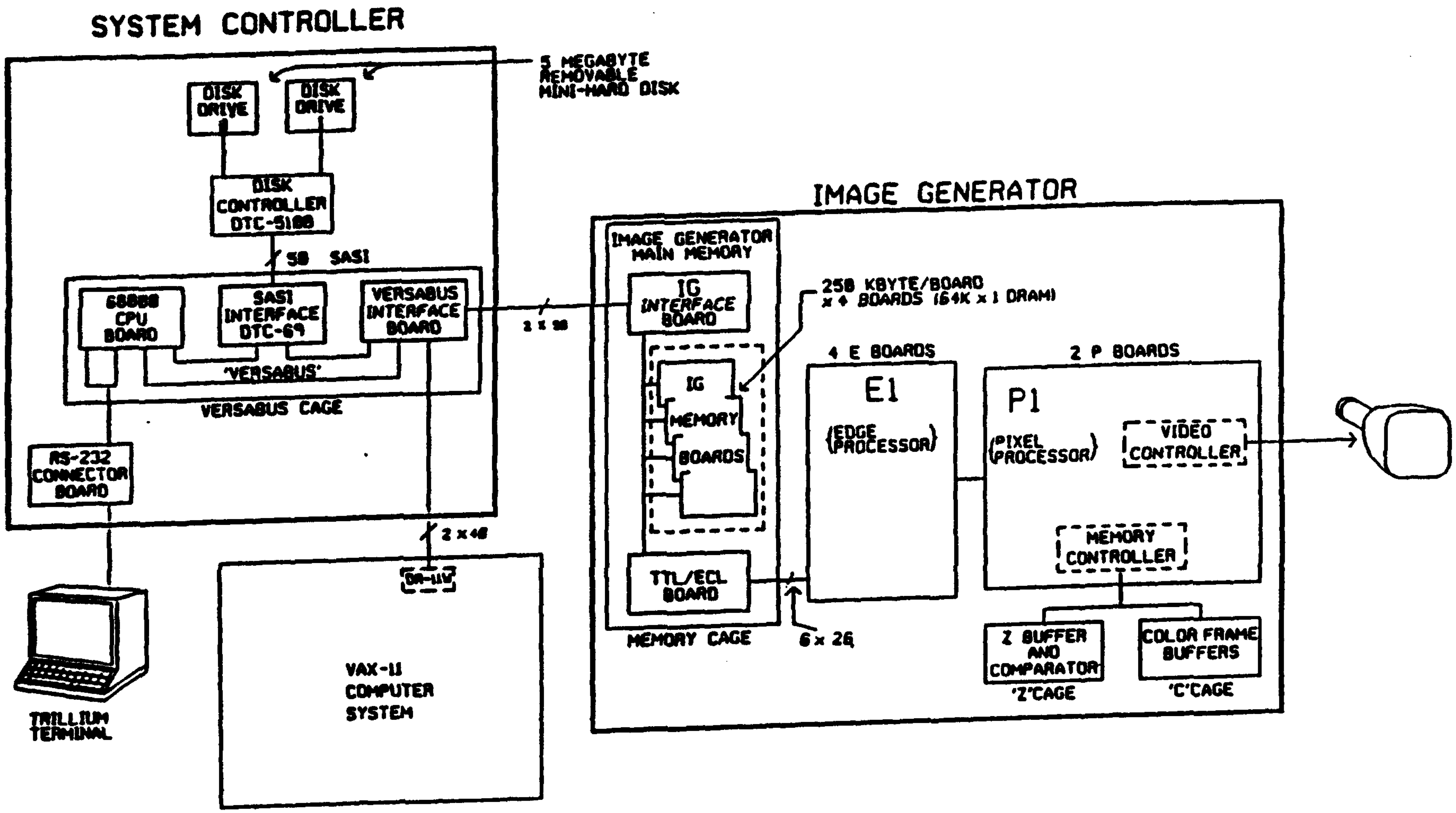


FIGURE 6. THE TRILLIUM 1102 PROCESSOR

The Trillium can image a 5000-polygon scene at the required 30 Hz rate. If the number of polygons exceeds this estimated limit, the update rate will slow down.

5.3 CHANNEL DATA BASE TRANSFORMATION

Section 4 describes the CDB and the filtering program which creates the three "child" data bases for the three sensor channels from the "parent" CDB for a new data base. Once the child data bases have been created, additional transformation (translation) is required to transform them into formats that are accepted by the Trillium and Graphicon hardware.

The transformation operations for the Trillium channels convert the CDB visual and FLIR data bases into a form that can be stored on the Trillium disks and then reloaded for a simulation run. The transformation consists of executing atom, object, and world compiler programs which generate a world binary file for each channel. This binary file contains all the imaging information for the scenario, and is structured to allow fast access to the graphics data base during real time execution. The world binary files are transferred to the Trillium via the VAX-Trillium DR-11W link. Once on the Trillium, the world binary files can be saved on the Trillium disks, using a Trillium utility program.

This transformation procedure is carried out only when a new data base is created.

5.4 REAL-TIME SIMULATION

When CDASS is started up for a simulation run, the selected simulation data base is loaded into the Trillium from the cartridge disks. Once the simulation is running, the Trillium accepts updates from the VAX and updates the images displayed on the visual and FLIR monitors.

The principal differences between the visual and FLIR channels are the orientation of the displays and the color. The visual channel orientation is fixed to the aircraft body, as is required of an out-the-window display. The FLIR channel is stabilized, and always displays a horizontal horizon with respect to the earth. In addition, there is a FLIR joystick which allows the pilot to rotate the FLIR sensor about a stabilized vertical axis and a stabilized horizontal axis.

The visual channel color is three-color, 27-bit RGB, which provides an accurate representation of colors. The FLIR channel has the same color capability, but the FLIR data base contains only monochrome shades in which red, green, and blue have the same intensity.

Communication between the master microVAX and the Trillium takes place over the high-speed parallel interface. The communication software consists of the Trillium-provided subroutines "initialize" and "send."

5.5 RELATIONSHIP OF TRILLIUM AND CDASS DATA BASES

Section 4 stated that the CDB format was derived from the Trillium data base format. The CDB format is simpler; it limits the atom file descriptions to four types: point definition by coordinate, normal definition by coordinate, polygon definition by point number, and vertex normal associations. This method of describing atoms is longer, but it simplifies manual editing and interpretation of atom files.

The way in which points and normals are described in the Trillium language can be quite elaborate. Points may be described individually by their x, y, and z coordinate values, or similarly they may be described as a list of points. Additional groups of points may be described by a transformation being applied to one of the lists of points. A number of transformations which may be applied, including: x, y, and z translation, scaling, and

rotation; general matrix operations; reflection along the x, y, and z axes; and an affine operator that maps the first four specified point coordinates to the last four specified. A list may be defined to describe a circular section. Lastly, unrelated points may also be grouped together as a list of points. Normals may be described by an x, y, and z component defining a direction towards which the normal points from the origin. Additionally, a normal may be described by listing points that define a surface. In this case the normal is automatically calculated by the atom compiler. Part two of the atom description contains the surface information of the atom. Here again, the way in which surfaces may be described in the Trillium language can be quite elaborate. Polygons may be described by a list of points that define the boundary of the polygon. Contours may be defined by associating lists of points with other lists of points or a single point. Contours may be Closed or Open. A Closed Contour is one in which the first and last points in each list are connected. Surfaces may also be described by moving a list of points one unit in the negative direction along any of the axes. Surface normals for contours may be described in which each point of a surface is assigned a normal. Finally, individual vertices may be associated with individual normals.

6.0 THE SAR CHANNEL

6.1 INTRODUCTION

The processing for SAR channel of CDASS differs significantly from the visual and FLIR channel. The SAR image is not displayed in real time and is implemented on a different graphics processor. More significantly, the SAR image requires the simulation of radar shadows, which have no equivalent in the visual and FLIR channels.

6.1.1 The SAR Image

The visual sensor (the pilot's eye) and the FLIR sensor form images by a 2-D projection of the scene in the field of view (FOV). The SAR sensor image is formed in range and azimuth and is usually displayed as a plan view of the scene. Thus the SAR image appears to be formed by a sensor directly above the patch being imaged, although the sensor is actually at some ground range away from the patch. Because the sensor is at a slant angle, the SAR image has radar shadows which are cast by any object between the radar and the ground. These shadows depend both on the scenario terrain and on the position of the aircraft.

The simulation of radar shadows requires both off-line data base processing and on-line processing. In the off-line processing, data base surfaces which could cast shadows are identified, classified, and placed into two shadow data base files. In the on-line processing, the shadow polygons are computed by a special algorithm optimized for speed and loaded into the Graphicon for image generation.

6.1.2 SAR User Interface

The area imaged by the SAR sensor can be controlled by the user, simulating the performance of an actual SAR sensor. The user can choose two modes: real beam ground map (RBGM) and SAR. In the RBGM mode, the display shows a 120-degree azimuth sector of the data base in front of the aircraft. The maximum range for the RBGM display is selected by the user; it can be either 25 or 45 nautical miles. All SAR displays are shown at the maximum resolution of the Graphicon; in particular, the degradation of imaging resolution in RBGM mode relative to the SAR mode is not simulated. Also, the SAR image can be generated with high resolution at all azimuths relative to the aircraft heading, which ignores the azimuth limitations of a real SAR sensor.

In the SAR mode, the user can select the location of the SAR patch and its size. The displayed patch is square, and the selectable sizes are 10, 5, 1, 0.2, and 0.05 nmi on a side. The displayed SAR image can be oriented in three ways: north up, radar line-of-sight up, or aircraft heading up.

6.2 THE GRAPHICON PROCESSOR

The Graphicon 1700 is a 3-D image generator, used to perform the display processing for VAX and Sun computers. (See Figure 7) Unlike the Trillium, the Graphicon was not designed to update its display at real-time speeds. Also, the Graphicon is limited to displaying a maximum of 4096 colors at one time. These 4096 colors can be selected from a palette of 16 million colors and loaded for each run. The data structure which specifies the displayed colors is called the video look-up table (VLT). The resolution of the Graphicon display is 1280 by 1024 pixels.

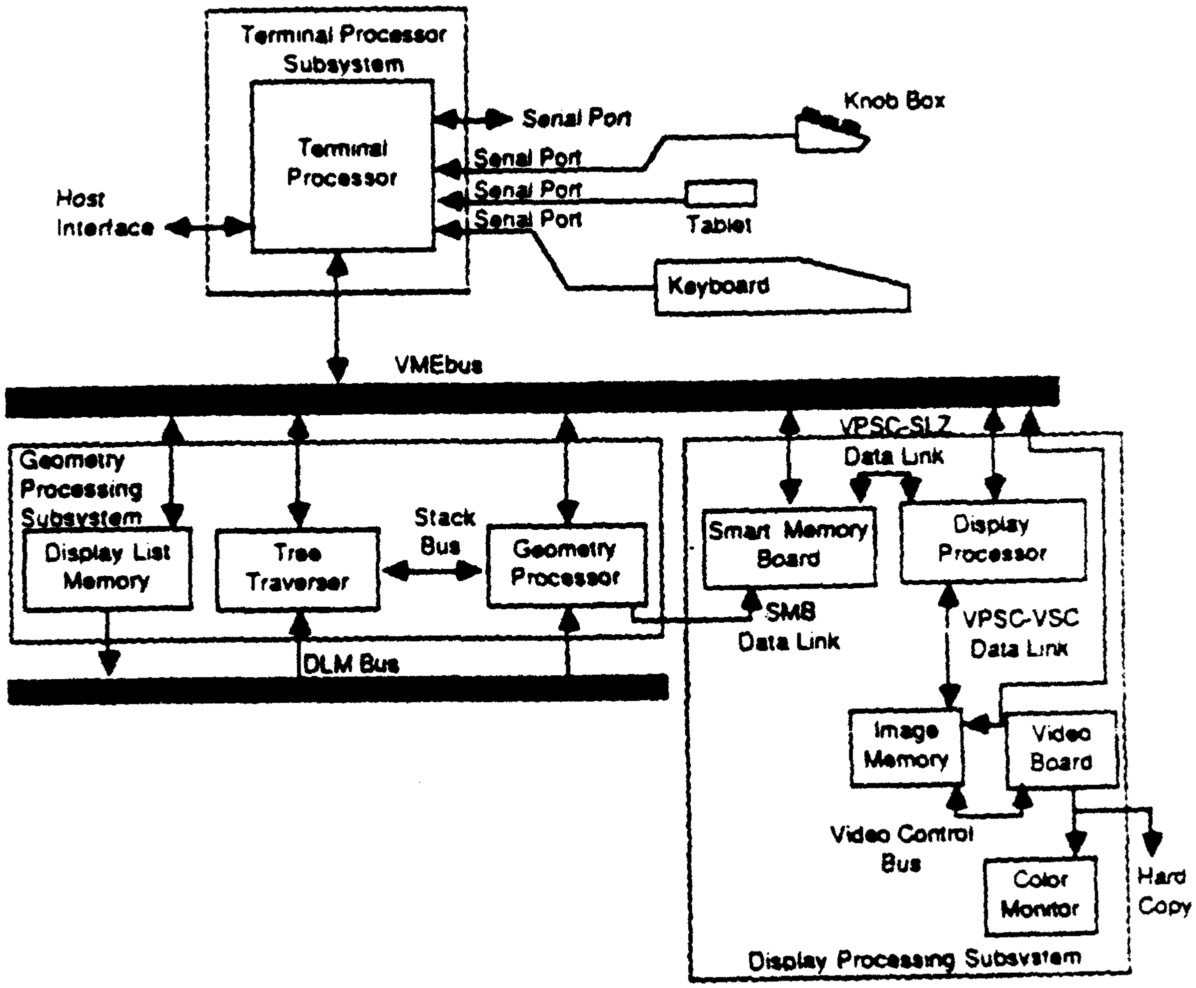


FIGURE 7. THE GRAPHICON 1700 PROCESSOR

6.3 SAR CHANNEL DATA BASE TRANSFORMATION

Section 4 describes the CDB and the filtering program which creates the three "child" data bases for the three sensor channels from the "parent" CDB for a new data base. Once the SAR child data base has been created, two additional operations are required to create a run-time data base: conversion to Graphicon format and shadow data base creation. Both of these procedures are carried out only when a new data base is created.

6.3.1 Graphicon Format Conversion

Graphics data is specified and loaded into the Graphicon using the Graphics Subroutine Library (GSL) provided by Graphicon and residing on the slave microVAX. The CDB child data base is first converted to the GSL format before loading into the Graphicon. Like the Trillium, the Graphicon stores current scenario data in a binary representation in its local memory. However, the Graphicon does not have a local storage device, so the SAR data bases are stored on the hard disk of the slave microVAX. At run time, the SAR data bases are loaded into the Graphicon.

6.3.2 The Shadow Data Bases

The CDB specifies the objects to be imaged by the SAR sensor. However, the SAR image includes shadows from objects in the FOV which must be explicitly inserted into the data base. (At the price-performance level of the Graphicon or Trillium, there is no automatic shadow generation.) Shadows are represented by horizontal black polygons, and are computed at run time from the shadow data base.

There are two major steps in the generation of the shadow data base. First, the shadow-producing polygons are identified and placed into a shadow polygon data base file. Figure 8 shows an example of a shadow-producing polygon, and the shadow polygon which will be generated in real-time. To produce the shadow polygon data base, every polygon in the SAR data base is tested for height above ground level (assumed to be at $Z = 0$). Depending on their height, polygons are classified as major shadowers, minor shadowers, or nonshadowers. The criterion for classification can be set by the user. Major and minor shadowers are placed in the shadow polygon file.

In the second off-line step, the shadow polygons are classified by their location in the data base. To reduce the real-time execution speed, it is important to compute only those shadow polygons that are in the neighborhood of the selected SAR patch. To facilitate this process, the data base is divided into a grid, and the shadow-producing polygons are classified by the grid cell in which they are located. This is done by overlaying a square grid on the entire data base region, dividing it into square cells or tiles. A file is created which specifies, for each tile, a list of the shadow-producing polygons in that tile. The shadow-producing polygon (.SHP) file and the tile data base (.TLB) file become a part of the SAR data base, to be used during the real-time processing.

6.3.3 SAR Reflectivity

The CDB object files contain a SAR reflectivity index, ranging from 1 to 16. This index is used to determine the display intensity of each of the data base surfaces which are visible in the SAR channel. The 16 values of the reflectivity index correspond to 16 different radar reflecting materials, whose reflective properties are simulated by CDASS.

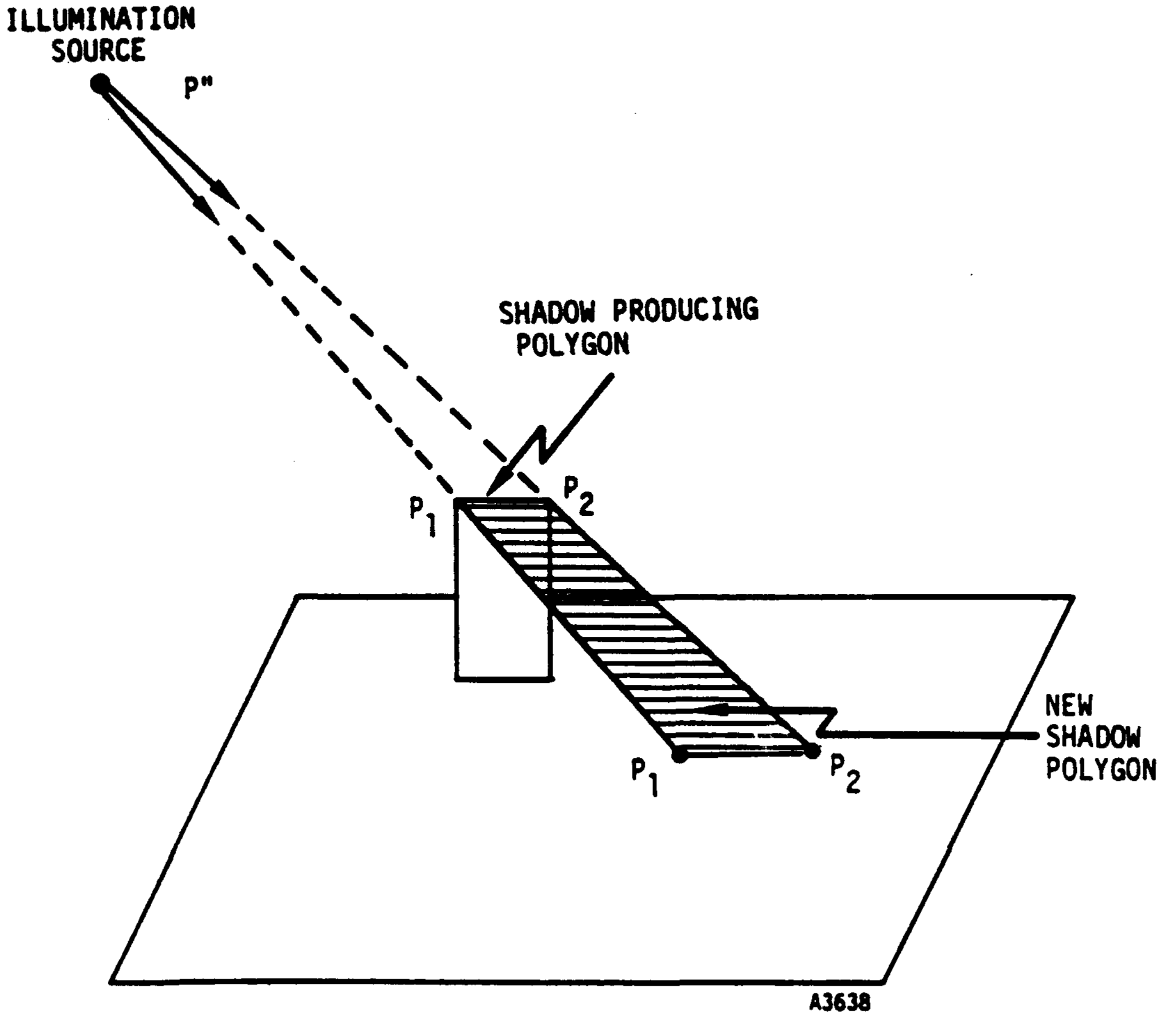


FIGURE 8. NEW SHADOW ALGORITHM

For each material, the index is used to look up the specular and diffuse reflectivity of the surface in a table. These reflectivities are then used to compute the reflected intensity of the surface as a function of angle for 256 angles between zero and 90 degrees. The set of 256 reflectivities is then loaded into the Graphicon VLT and used to render the surface, as shown in Figure 9. As part of its real-time imaging computations, the Graphicon computes the line-of-sight angle to the surface and looks up the correct intensity for that angle. The limitation to 16 materials is due to the size of the Graphicon VLT: 16 materials times 256 angles is 4096 entries.

The table of material reflectivities can be modified by the user and the limit of 16 different materials only applies within one data base. Potential CDASS users should clearly understand the method by which CDASS simulates SAR images before they begin to create SAR data bases. The appearance of a object in the SAR channel is computed from a plan view image and therefore depends on the horizontal projection of its surfaces. However, the true SAR image is formed by reflection of the radar echo at the true sensor angle, which is usually closer to horizontal than to a plan view. Therefore, the radar reflectivity assigned to the roof of a rectangular building will be used to generate the SAR image; the radar reflectivity assigned to the walls will have no effect because the walls are not visible in a plan view. However, the true SAR image will depend on the reflectivity of the walls. Therefore, the user must assign a radar reflectivity to the roof which simulates the total SAR return from the building.

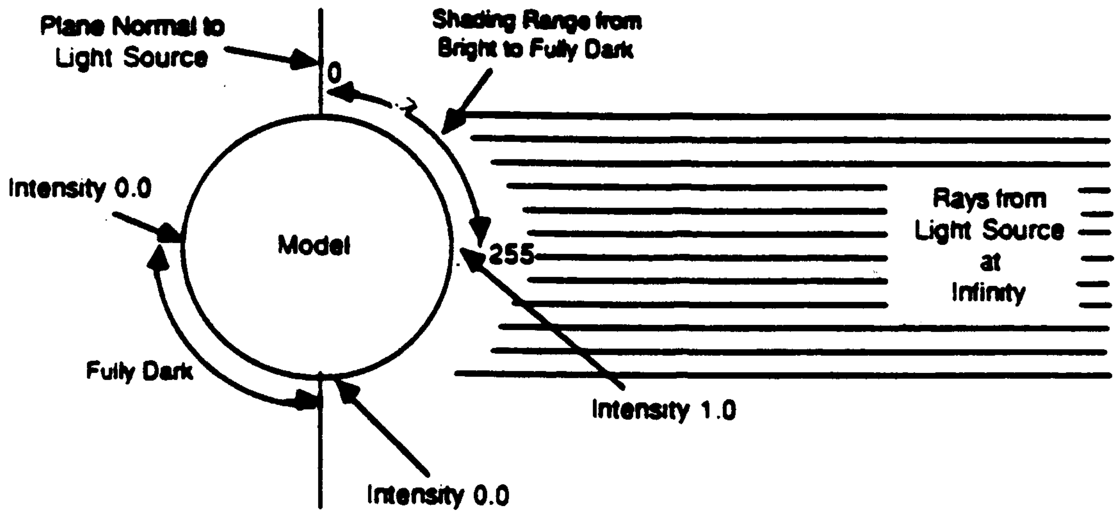


FIGURE 9. GRAPHICON SURFACE SHADING

6.4 SAR IMAGE SIMULATION

When CDASS is started up for a simulation run, the SAR data base is loaded into the Graphicon memory. Whenever the user requests an RBGM or SAR image, the slave microVAX computes the vertex coordinates of the shadow polygons, using the shadow data base files and the current aircraft position. The computed shadow polygons are then loaded into the Graphicon display memory and the complete SAR data base (normal polygons and shadow polygons) is displayed as the SAR image.

To avoid computing the shadow polygon for every shadow-producing polygon in the data base, the tile data base is used. The geometry of the tile data base and a SAR patch are shown in Figure 10. The real-time program extracts only the tiles that could affect the image, and the shadow-producing polygons that lie within those tiles. For tiles that intersect the SAR patch (eight tiles in Figure 10), shadows are generated for both major and minor shadow-producing polygons.

However, there may be shadows in the SAR patch from tiles that do not intersect the SAR patch but lie within the line-of-sight of the radar to the SAR patch (seven tiles in Figure 10). For this latter set of tiles, shadows are generated for only the major shadow-producing polygons. The major shadow-producing polygons, such as mountains, are sufficiently high that their shadows may extend into other tiles.

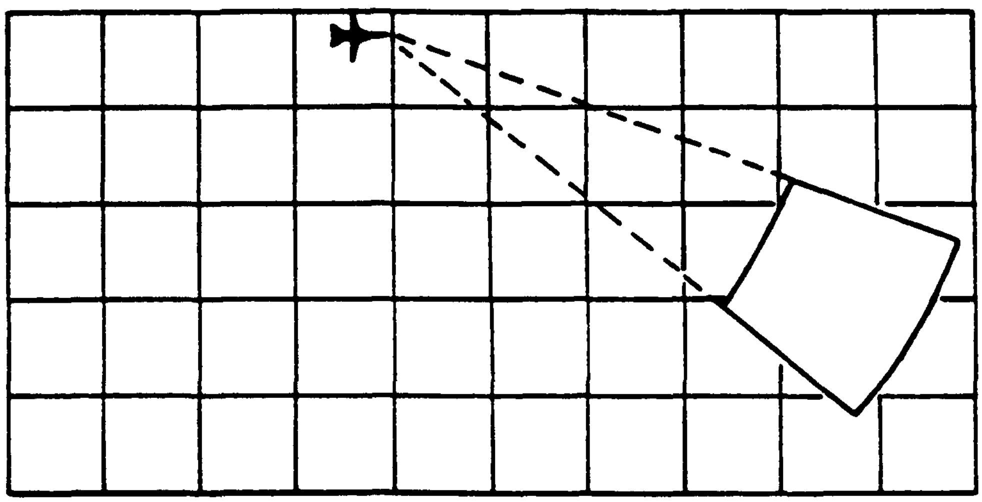


FIGURE 10. TILE DESCRIPTION LISTS

7.0 MODES OF OPERATION

7.1 HOST INTERFACE MODE

When CDASS is functioning as a unit within the AVSAIL simulation system, it receives its commands from the Harris simulation computer and operates synchronously with the other simulation displays. This mode of operation is known as the host interface mode. The Harris is connected to the master microVAX by a DR-11W high-speed parallel interface designed by AVSAIL.

In the host interface mode, CDASS receives an aircraft position and orientation update, and uses the updated information to update the visual and FLIR displays. The host update and Trillium update are asynchronous and independent.

7.2 STAND-ALONE MODE

The stand-alone mode allows CDASS to operate as an independent three-channel simulator. In this mode, CDASS accepts inputs from the pilot interface and predicts the aircraft state using its own motion model.

7.2.1 Pilot Interface

The pilot interface uses two joystick and the master microVAX terminal. One joystick is used to control aircraft attitude, with the back-and-forth motion controlling pitch, and the left-right motion controlling combined bank and turn. The FLIR joystick, which determines the pointing azimuth of the FLIR, can also be used to vary aircraft speed.

The terminal keyboard can be used to start and stop aircraft motion, and while stopped, can be used to reposition the aircraft in X, Y, altitude, speed, heading, pitch, and bank. The terminal display provides a continuous

readout of these quantities. The contents of the pilot's display are shown in Figure 11.

7.2.2 The Motion Model

The motion model predicts aircraft motion and determines the aircraft response to the user's joystick control inputs. The current version of CDASS implements the motion model for a jet trainer, but there is a provision for including motion models for other types of aircraft.

The motion model integrates the equations of motion using a simple six-degree-of-freedom dynamics model. The three translational equation include only the forces of lift and gravity, assuming a balance between drag and thrust. The lift coefficient is tabulated as a function of the angle of attack. The rotational equations are integrated using the pitch and roll rates which are read from the joysticks. A zero yaw angle of attack is assumed.

```
INTERACTIVE FLIGHT MENU          |              TIME:    0:02
                                 |
 1. RESUME                       |       AIRCRAFT                    SAR
 2. FREEZE                       |   (database coord)   HOLD   (database coord)
 3. SAR Orientation              |
 4. X                            |         X:  -0.8 NM          CENTER OF PATCH:
 5. Y                            |         Y:  -7.9 NM                X:   0.0 NM
 6. ALTITUDE                     |       ALT:   699 FT                Y:   0.0 NM
 7. HEADING                      |   HEADING:     1 DEG
 8. PITCH                        |     PITCH:    -3 DEG           (aircraft coord)
 9. BANK                         |      BANK:   -48 DEG        GROUND RANGE:  0.0 NM
10. SPEED                        |     SPEED:   241 KNOTS           AZIMUTH:    0 DEG
11. SELECT MARK POINT Menu       |  CLIMB RT:   -25 FT/SEC
12. RESET to MARK POINT          |
13. RECORD                       |
14. RETURN to CDASS Mode Menu    |         FLIR                 PROGRAM OPERATIONS
---------------------------------|
                                 | POINTING ANGLE: L 45 DEG     MODE: INTERACTIVE
  PLEASE ENTER THE COMMAND       |      DEP ANGLE: , 36 DEG     OPERATIONS: RUNNING
          NUMBER:                |                              RECORDING: OFF
                                 |                              AIRCRAFT MODEL: JLI
---------------------------------
```

A6250

FIGURE 11. INTERACTIVE FLIGHT MENU

8.0 CONCLUSIONS

The CDASS project ended with the successful development and installation of the three-channel flight simulator and the common data base development environment. The project therefore validated the basic concepts of low-cost multi-channel simulation and a multi-sensor data base. Many significant lessons were learned during the CDASS program; this section discusses these lessons learned and their application to future flight simulator development.

8.1 CDASS IMAGE GENERATION HARDWARE

During the four years that elapsed between the CDASS proposal and the installation of the finished system, significant advances were made in the technology of real-time computer graphics. A moderate-performance system which cost about $1 million in 1985 is now available for about one-tenth of that price. There is no doubt that if CDASS were being designed in late 1989, hardware performance would be superior and hardware costs would be far less than for the delivered 1987-vintage system.

This rapid and inevitable obsolescence of computer hardware justifies the CDASS concept of a device-independent data base, which can be hosted on future graphics hardware. It also suggests that periodic upgrades of the CDASS hardware are necessary if state-of-the-art performance is to be maintained. Therefore, it would be desirable to buy CIG hardware from one of the major companies specializing in the real-time graphics area; this would minimize major modifications as the product line is upgraded.

8.2 REAL-TIME PERFORMANCE

The real-time system met both the contract requirements and user expectations for the visible and FLIR channel. A 30-Hertz update rate was

maintained for all scenes within the AVSAIL terrain belt data base. The displayed images occasionally showed artifacts due to the limited z-buffer resolution of the Trillium; these artifacts were visible when two closely spaced planes were viewed from a long range.

In the SAR channel, the update time was on the order of 5 to 10 seconds, which exceeded user expectations. Most of this update time is due to the computation of the shadow polygons, which takes place on the microVAX. This update time could be reduced by optimization of the shadow generation algorithms, and/or by replacing the MicroVAX II with the current MicroVAX III.

8.3 THE COMMON DATA BASE

The concept of a device-independent multi-sensor common data base proved to be feasible and not too difficult to implement. Translators from the common data base to Trillium and Graphicon formats were written, and writing translators for other CIG hosts should be straightforward. The tools written for creating common data bases performed satisfactorily, although major improvements could be made in this area. The object modeler (AutoCAD) and World Editor run on different machines and require several manual operations to transfer and convert files. The World Editor does not perform at the level of the best commercial data base creation tools such as MultiGen, a product of Software Systems which runs on the Silicon Graphics workstations. However, these commercial tools are the result of a considerably greater investment than the six man-months that went into the World Editor and its documentation.

During the CDASS development period, there were no off-the-shelf data base creation tools for either the Trillium or Graphicon. In choosing future CDASS graphics hardware, the availability of data base modeling tools to replace the object modeler and World Editor should be a factor. As of late 1989,

Graphicon was offering its version of MultiGen (called TopGen) for its 1700S and 2000 real-time processors.

Finally, it would be desirable to have a more automated data base management system, which would catalog the data bases, perform the required translations and transformations, and keep track of versions and updates.

8.4 TERRAIN MODELS

The CDASS experience with creating and flying terrain models indicate that for an aircraft flight simulator, the most important qualities are (1) good models of terrain surfaces, (2) terrain surface texture, and (3) data base population with cultural objects. Details in individual cultural objects was not found to be significant.

The CDASS data base and graphics hardware are polygon-based (as are all current real-time graphics systems), and experience suggests that considerable effort should be invested in creating detailed topographical contours. A crude polygonal model of a hill, even with Gouraud shading, detracts from the realism of the simulator. This also emphasizes the need for a good DMA conversion capability.

Terrain surface texture requires special hardware which was not available within the CDASS cost constraints. Beginning in 1990, low-cost textured scene generators (such as the Graphicon 2000) will become available and should be considered for any future CDASS upgrade.

The common data base format is well-suited for populating cultural areas because of its hierarchical structure. For example, a group of houses can be represented as one object and easily duplicated throughout the data base.

8.5 SENSOR MODELS

One of the applications of CDASS is the evaluation of airborne sensors. However, the current CDASS system has a limited capability for modeling the IR and radar sensors. If either sensor were modified, for example, to have a greater sensitivity or dynamic range, the CDASS user would have to manually compute and reassign gray shades to every surface in the data base, and would have to modify the module which assigns radar reflectivities to the surfaces.

A future upgrade to CDASS would include sensor models which would compute the displayed surface color from the surface material type and the characteristics of the sensor. The displayed data base could then be updated automatically as sensor characteristics were changed.

8.6 DMA CONVERSION

As mentioned in section 8.4, good terrain representation and cultural object population are key elements in flight simulation. A capability to convert DMA digitized terrain elevation data (DTED) and digitized cultural features to CDB format would be a very significant asset for the CDASS system.

The Trillium Corporation provided two versions of a program to convert DMA DTED data to Trillium binary data base files. The program creates polygons representing the terrain surface from the digitized elevation grid. It averages (filters) the resulting grid to reduce the number of polygons in the model. The Trillium program was tested and found to be inadequate; the filtering was not sufficiently sophisticated and the resulting terrain was not a good representation of the DTED data. In addition, the program was not robust, and additional modifications would have to be made to convert the output from the Trillium binary data base to the CDB format.

Since there are existing DMA DTED conversion programs, it may be worthwhile to adapt them to provide output in the CDB format.

☆U.S. GOVERNMENT PRINTING OFFICE: 1992 - 648-127/62386